# Program Assignment Recursive Descent Parser

Compiler Theory #1
21600193
Hyo-Rim, Kim
May 13th, 2019

#### 1. EBNF

```
program \rightarrow stmt-sequence | \epsilon
stmt-sequence → statement {; stmt-sequence}
statement →for-stmt | while-stmt | if-stmt | class-stmt | func-stmt | assign-stmt | main-stmt | else-stmt
main-stmt \rightarrow main() \{ stmt-sequence \}
if-stmt \rightarrow if (expr) {stmt-sequence}
else-stmt \rightarrow else {stmt-sequence} | if-stmt
class-stmt → class id {stmt-sequence}
func-stmt \rightarrow id(func-parameter)
assign-stmt \rightarrow id = expr | id++ | id- -
for-stmt \rightarrow for(assign-stmt; expr; assign-stmt){stmt-sequence}
while-stmt \rightarrow while(expr){stmt-sequence}
expr \rightarrow < term > {< addop > < term >} | < term > {< cmpop > < term >}
func-parameter \rightarrow id \{, id\} | literal
term \rightarrow <factor> {< mulop> < factor>}
addop \rightarrow + | -
cmpop \longrightarrow <|<=|>|>=|==|!=
mulop \longrightarrow * \mid / \mid \%
factor \rightarrow number | id | (expr)
```

# 2. Design Document

#### 1. UML

# scnr +public static int keyflag +public static int str\_length +public static int idx\_str +public static String cnt\_str +public static String cur\_str +public static char cur +public static char row +public static char col +public static int state +public static char before +public static char after +public static int tok1 +public static int tok2 +public static int expr\_flag + main(String[] args): static void + stmt\_sequence(): static int + get\_token\_one(): static int + statement(): static int + main\_stmt(): static int + if\_stmt(): static int + else\_stmt(): static int + class\_stmt(): static int + func\_stmt(): static int + assign\_stmt():static int + for\_stmt():static int + while\_stmt():static int + expr():static int + func\_parameter():static int + term():static int

+ get\_token():static int[]

```
+ readFile(final String fileName): static String
```

+ get\_token\_type(String str, int state): static int

#### 3. Result

## 1. Use Java

```
class : keyword
MyClass : id
{ : left curl
  : left curly brace
main : keyword
( : left parenthesis
) : right parenthesis
{ : left curly brace
int : keyword
$_TimeO : id
  : assignment symbol
22 : number literal
 : semicolon
; : semicolor
if : keyword
( : left parenthesis
$_TimeO : id
< : greater than symbol
10 : number literal
) : right parenthesis
{ : left curly brace
out.println : keyword
( : left parenthesis
" : double quote symbol
Good : literal
morning. : literal
  : double quote symbol
) : right parenthesis
; : semicolon
} : right curly brace
else : keyword
if : keyword
( : left parenthesis
$_TimeO : id
  : greater than symbol
20 : number literal
) : right parenthesis
{ : left curly brace
out.println : keyword
( : left parenthesis
" : double quote symbol
Good : literal
  : double quote symbol
) : right parenthesis
; : semicolon
} : right curly brace
else : keyword
{ : left curly brace
out.println : keyword
( : left parenthesis
" : double quote symbol
Good : literal
evening. : literal
  : double quote symbol
) : right parenthesis
  : semicolon
} : right curly brace
  : right curly brace
 : right curly brace
Parsing Ok
 nyorm@ariselab:~/cmpr/hw2_21600193/use_javac/src$
```

#### 2. Use Ant

```
build:
run:
       [java] class : keyword
       [java] MyClass : id
[java] { : left cur
                { : left curly brace
       [java] main : keyword
       [java] ( : left parenthesis
[java] ) : right parenthesis
               { : left curly brace
       [java]
                int : keyword
       [java]
       [java] $_TimeO : id
       [java] = : assignment symbol
       [java] 22 : number literal
[java] ; : semicolon
                if : keyword
       [java]
       [java] ( : left parenthesis
[java] $_TimeO : id
                < : greater than symbol</pre>
       [java]
                10 : number literal
       [java]
       [java] ) : right parenthesis
[java] { : left curly brace
       [java] out.println : keyword
                ( : left parenthesis
" : double quote symbol
       [java]
       [java]
       [java] Good : literal
       [java] morning. : literal
       [java]
                   : double quote symbol
                ) : right parenthesis
       [java]
       [java] ; : semicolon
[java] } : right curly brace
[java] else : keyword
[java] if : keyword
       [java] ( : left parenthesis
[java] $_TimeO : id
       [java]
               < : greater than symbol</pre>
                20 : number literal
       [java]
       [java] ) : right parenthesis
[java] { : left curly brace
               out.println : keyword
       [java]
       [java] ( : left parenthesis
[java] " : double quote symbol
       [java]
       [java] Good : literal
       [java] day. : literal
       [java]
                   : double quote symbol
                ) : right parenthesis
       [java]
       [java] ; : semicolon
[java] } : right curly brace
       [java] else : keyword
[java] { : left curly brace
       [java] out.println : keyword
       [java] ( : left parenthesis
[java] " : double quote symbol
       [java] Good : literal
       [java] evening. : literal
                   : double quote symbol
       [java]
       [java] ) : right parenthesis
       [java] ; : semicolon
       [java] } : semicoron
[java] } : right curly brace
[java] } : right curly brace
[java] } : right curly brace
[java] Parsing Ok
BUILD SUCCESSFUL
Total time: 1 second
```

#### 5. User Manual

- 1. Use ant (directory name)
  - 1. ant version
    - Apache Ant(TM) version 1.10.5 compiled on March 28 2019

## 2. build.xml

```
cproject name="scnr_prsr" default="build" basedir=".">
       cproperty name="src" value="src"/>
       cproperty name="build" value="build"/>
       cproperty name="doc" value="doc"/>
       <path id="lib.path">
              <pathelement location="${build}" />
       </path>
       <target name="init">
              <mkdir dir="${build}"/>
       </target>
       <target name="build" depends="init">
              <javac srcdir="${src}" destdir="${build}" debug="true"</pre>
includeantruntime="false">
              </javac>
       </target>
       <target name="run" depends="build">
              <java classname="scnr_prsr" fork="true" dir="." maxmemory="4096m">
                     <classpath location="."/>
                     <classpath refid="lib.path"/>
                     <arg file="data/test.txt"/>
              </java>
       </target>
       <target name="clean">
              <delete dir="${build}"/>
```

```
</project>
```

## 6. command

- ant build
- ant run
  - this build.xml already set the file name(test.txt)

# 7. Use Javac (directory name)

- 1. java version
  - openjdk version "11.0.6" 2020-01-14
  - OpenJDK Runtime Environment (build 11.0.6+10-post-Ubuntu-1ubuntu118.04.1)
  - OpenJDK 64-Bit Server VM (build 11.0.6+10-post-Ubuntu-1ubuntu118.04.1, mixed mode)

#### 2. command

- javac scnr.java
- java scnr [file name]